

P5285a

APPLICATION

FOR

UNITED STATES LETTERS PATENT

Be it known that we, Yuji Takamizawa, and Atsushi Yamaji, both citizens of Japan, of 3-5 Owa 3-chome, Suwa-shi, Nagano-ken, 392 Japan, c/o Seiko Epson Corporation, have invented new and useful improvements in:

PRINTER AND CONTROL METHOD FOR THE SAME

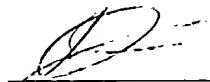
of which the following is the specification.

CERTIFICATION UNDER 37 C.F.R. 1.10

"Express Mail" Mailing Label Number: EL700475059US

Date of Deposit: October 16, 2000

I hereby certify that this patent application is being deposited with the United States Postal Service on this date in an envelope as "Express Mail Post Office to Addressee" service under 37 C.F.R. 1.10 on the date indicated above and is addressed to the Assistant Commissioner for Patents, Washington, DC 20231.



Denise Wendler

009T0T" 46906960

PRINTER AND CONTROL METHOD FOR THE SAME

Inventors: Yuji Takamizawa
Atsushi Yamaji

BACKGROUND OF THE INVENTION**5 Field of the Invention**

5.1 A1 > ~~The present invention relates to a printer that operates in response to data and control commands received from a host computer, and relates to a control method for such printer. More particularly, our invention relates to such a printer and a method for controlling the printer when the printer is off-line.~~

10 Description of the Related Art

Printers connected to a host computer receive data, including control commands, from the host and operate to print as instructed by the received control commands. Received data is temporarily stored in a first-in, first-out (FIFO) receive buffer, and sequentially interpreted therefrom by the analyzer section of the CPU. 15 The printer then performs a printing process according to interpreted control command content.

Real-time commands can be contained in the received data. Real-time commands are control commands that are run immediately upon receipt by the printer without being stored to the receive buffer. When a real-time command is received, the printer inserts the real-time command at the front of the command queue for immediate execution before other stored commands. Real-time commands include, for example, a command to return information about the printer status to the host, and commands for recovering the printer from recoverable errors. These commands can be performed by the printer even when an error has occurred and 25 the printer is off-line.

Printers of this type are normally either off-line or on-line. When on-line, data received from the host computer is stored to the receive buffer and sequentially interpreted therefrom by the CPU.

When off-line, data in the receive buffer is not interpreted. Conventionally, 30 when the printer is off-line data can still be received from the host computer and sequentially stored to the receive buffer. The CPU, however, interprets only real-time commands and does not interpret data in the receive buffer when off-line.

This means that the receive buffer can become full (meaning that there is either very little or no space left to store more data) when a large volume of data, such as from a bit image, is received from the host when the printer is off-line. The communication interface of the printer then enters a busy state, indicating that no more data can be received from the host, and thus prevents any additional data, including real-time commands, from being sent from the host to the printer. The host is then unable to detect the printer status or recover the printer from recoverable errors.

Referring to Fig. 5, for example, the receive buffer 50 of the printer obviously has a finite capacity, which in this example is shown to be 4096 bytes. The printer indicates a busy state to the host computer whenever a certain specific amount of data is stored to the receive buffer 50. In this example a busy state is indicated when the receive buffer is filled to at least 95% of its capacity (i.e., when 3840 bytes have been stored in this example). If the printer has a print head that prints 640 dots per horizontal line (80 bytes per line), the receive buffer will become full when data sufficient to print 48 dot lines (about 6 mm vertically across the page) has been received ($3840/80 = 48$ dot lines).

This means that if bit map print data such as shown in Fig. 6 equivalent to about one typical receipt is received while the printer is off-line, the receive buffer will be filled almost immediately.

If the host computer sends data to the printer via a print spooler and the printer keeps sending a busy signal, the print spooler will eventually generate an error. However, the application that has sent the data to the spooler cannot delete data from the spooler, and the user must therefore directly delete the unprinted data from the spooler.

Objects of the Invention

Having identified these problems of a conventional printer, our invention provides a printer control method and a printer in which the receive buffer does not become full even when a large volume of data is received while the printer is off-line, and which therefore prevents the communication interface with the host computer from entering a busy state.

A further object of our invention is to provide a printer control method and printer that can select a mode for storing or a mode for destroying received data when the printer is off-line.

A yet further object of our invention is to provide a printer control method and printer that interprets real-time commands and performs a process according to the content of the real-time command when the printer is in a mode for avoiding a busy state.

5 A yet further object of our invention is to provide a printer that can delete received data rather than the application program on the host computer deleting transferred data, and thus does not increase the load on the application program.

10 A yet further object of our invention is to assure that print data sent by the host computer can be reliably printed even if the printer is set to the mode for destroying received data.

Summary of the Invention

5.5 A2> To achieve these objects, a control method according to our invention is for a printer having a receive buffer for temporarily storing data including a control command from a host computer, and operating according to the interpreted content of data in the receive buffer. The control method has a step for detecting whether the printer is in an off-line state in which received data is not interpreted, or an on-line state in which received data is interpreted; and a clearing unit for implementing a step for clearing data in the receive buffer when the printer is detected to be in the off-line state.

20 The potential for the receive buffer to become full when the printer is off-line is thus reduced, and the interface with the host computer can be prevented from entering a busy state in most circumstances.

The receive buffer data clearing step in this case is preferably accomplished immediately after the printer goes off-line.

25 Yet further preferably, the control method additionally has a step for setting the method for handling data when the printer is off-line; and a step for reading this data handling method setting when the printer is detected to be off-line. In this case the receive buffer data clearing step can be controlled to run only when the read data handling method is set to allow clearing data in the receive buffer.

30 The printer user in this case can select whether to clear or save data received when the printer is off-line, and printer operation can thus be set according to how the printer is being used.

5.6 A2> Yet further preferably, the step for setting the off-line data handling method is accomplished according to a specific control command from the host computer. In this case the control bit, or word, setting is stored to memory in the printer

according to the control command from the host. Other methods can, of course, be used to set and save the data handling method setting, including DIP switches on the printer itself, or writing the data handling method to fuse ROM or flash memory.

- 5 Yet further preferably, the control method has a step for destroying data received from the host computer until the printer goes on-line after the step for clearing data in the receive buffer.

In this case there is no chance for the receive buffer to become full when the printer is off-line.

- 10 Further preferably, the step for destroying data received from the host computer is accomplished only when the off-line data handling method is set to allow destruction of data received from the host computer.

- 15 The control method can obviously also have in place of the step for destroying data a step for saving in the receive buffer data received from the host computer until the printer goes on-line after clearing data in the receive buffer.

Yet further preferably, this control method additionally has a step for detecting whether the printer is on-line or off-line after the printer is detected to be off-line; and a step for clearing data in the receive buffer when the printer is thus detected to be on-line.

- 20 Yet further preferably, the printer also has a print buffer for storing expanded print data, and the steps for clearing data in the receive buffer clear data in the receive buffer and clear data in the print buffer.

- 25 Yet further preferably, the printer also has a step for determining whether data received by the printer when off-line is a control command to be executed immediately, that is, a real-time command; a step for interpreting the control command and operating the printer according to the interpreted content when the received data is determined to be a real-time command; and a step for destroying said control command.

- 30 Control by means of real-time commands is thus enabled even when the printer is off-line, and the printer status can be polled and recoverable printer errors can be corrected using real-time commands.

- 35 Our invention also relates to a control method for a host computer for sending data including control commands to a printer operating according to any of the above noted control methods of our invention. This host computer control method has a step for sending print data to the printer in conjunction with a command

requesting notification of printing completion; a step for awaiting notification of printing completion from the printer in response to said request command; and a resending unit for implementing a step for resending print data to the printer after receiving an on-line notification from the printer when an off-line notification is received from the printer while awaiting printing completion notification. The resending unit may include a CPU with a program set stored in the ROM ("and a component Q" if needed).

Data sent from the host computer when the printer is off-line is destroyed in a printer controlled by a method of our invention, but by using this host computer control method the host computer can detect the printer process, that is, whether printing previously sent print data was completed, and can resend the print data if it has been destroyed.

The present invention also relates to a printer having a receive buffer for temporarily storing data including a control command from a host computer, and operating according to the interpreted content of data in the receive buffer. This printer has a line state detection means for detecting whether the printer is in an off-line state in which received data is not interpreted, or an on-line state in which received data is interpreted; and a data clearing unit, or means, for clearing data in the receive buffer when the printer is detected to be in the off-line state by the line state detection means.

This printer further preferably has a means for setting a method for handling data when the printer is off-line; and a data handling method reading means for reading this data handling method setting when the printer is detected to be off-line by the line state detection means. In this case the data clearing means clears the receive buffer only when the data handling method read by the reading means is set to allow clearing data in the receive buffer.

Yet further preferably, the means for setting the off-line data handling method sets the method according to a specific control command from the host computer.

Yet further preferably, the printer has a data destroying means for destroying data received from the host computer while the printer is off-line.

In this case the data destroying means destroys data received from the host computer only when the data handling method read by the data handling method reading means is set to allow destruction of data received from the host computer.

The data clearing means can be further controlled to clear data in the receive buffer even when the line state detecting means detects an on-line state after detecting an off-line state.

5 Yet further preferably, the printer additionally comprises a print buffer for storing expanded print data. In this case the data clearing means clears data in the receive buffer and clears data in the print buffer.

10 Yet further preferably, the printer also has a command interpreting means for determining whether data received by the printer when in an off-line state is a control command to be executing immediately, that is, is a real-time command; a printer operating means for interpreting the control command and operating the printer according to the interpreted content when the received data is thus determined to be a real-time command; and a command destroying means for deleting said control command.

15 Our invention also provides a printer having an interface for receiving data from a host computer; a receive buffer for temporarily storing data received by the interface; a storage means for storing an indicator of whether the printer is in a data clearing mode or a data saving mode when in an off-line state in which received data in the receive buffer is not interpreted; and a data clearing means for clearing data in the receive buffer when in the off-line state and the data clearing mode setting is stored by the storage means.

20 Further preferably in this case, the storage means sets the data clearing mode and data saving mode setting according to a command received from the host computer.

25 In this case the receive buffer is cleared even when the printer is off-line if the printer is set to the data clearing mode. The receive buffer therefore has no chance to become full, and real-time commands from the host computer can be easily received and processed.

30 Yet further preferably, the printer has a command execution means for running an executable command whether in the off-line state or on-line state, regardless of the mode stored by the storage means, when a command that is executable in an off-line state is received by the interface from the host computer.

35 Yet further preferably, the printer has a control means for controlling buffering data received from the host computer so that new received data is not stored to the receive buffer when the printer is off-line and the data clearing mode is set in the storage means. Alternatively in the clearing mode, data can be destroyed by continuing to store and process received data while the printing motor

is not driven such that no printing results from the interpreting, i.e. processing, the received data.

Thus comprised, data received from the host computer is destroyed, the receive buffer does not become full, and the printer can therefore receive and run real-time commands from the host computer.

Because the printer can receive real-time commands from the host computer, the cash drawer of a POS station can be opened even when the printer is off-line, for example. In addition, when the user rectifies an error due, for example, to a paper jam or paper cutter jam, causing the printer to go off-line, a real-time command for recovering from the error can be received from the host computer, and the printer can be brought back on-line. Furthermore, the printer can run various real-time status request commands sent by the host, and can thus return the printer status causing the error or off-line state and keep the host informed of the current printer status even while the printer is off-line.

Our invention yet further relates to a host computer for sending data and control commands to a printer as described above. This host computer has a data transmission means for sending print data accompanied by a command requesting printing completion notification to the printer; a notification detection means for awaiting and detecting a printing completion notification from the printer in response to the requesting command; a line state detection means for awaiting and detecting an on-line state or off-line state notification from the printer; and a print data resending means for resending print data to the printer after receiving an on-line notification from the printer when the line state detection means receives an off-line notification from the printer while awaiting printing completion notification.

Our invention provides a control method for a printer having a receive buffer for temporarily storing data including a control command from a host computer, and operating according to the interpreted content of data in the receive buffer. This control method has a step for detecting whether the printer is in an off-line state in which received data is not interpreted, or an on-line state in which received data is interpreted; and a step for destroying without storing to the receive buffer data received from the host computer when the printer is detected to be off-line.

This printer control method is also characterized by running only real-time commands in the data received from the host computer.

With this method the receive buffer will not become full when the printer is off-line, and a busy signal will therefore not be sent from the printer to the host computer.

The printer can therefore reliably receive and run real-time commands sent from the host computer.

Other objects and attainments together with a fuller understanding of the invention will become apparent and appreciated by referring to the following description and claims taken in conjunction with the accompanying drawings.

Brief Description of the Drawings

Fig. 1 is a schematic block diagram showing the general connections between a host computer and a printer according to a preferred embodiment of our invention;

Fig. 2 is a flow chart of the control process when the printer according to a preferred embodiment of our invention is off-line;

Fig. 3 is a flow chart of the control process of host computer operation when sending print data to the printer;

Fig. 4 is a flow chart of printer operation when print data is sent from the host computer in accordance with the control process shown in Fig. 3;

Fig. 5 is a schematic diagram of the receive buffer describing a receive buffer full state; and

Fig. 6 is an typical example of a bit map image sent to a printer for printing receipts.

Description of the Preferred Embodiments

The preferred embodiments of our invention are described next below with reference to the accompanying figures, of which Fig. 1 is a schematic diagram showing the general connections between a host computer and a printer of our invention.

As shown in Fig. 1, printer 15 is connected to a host computer 10 (referred to as simply the host below) by way of a Centronics (R) or other interface cable 14. Printer 15 receives control commands and other data from host 10, and operates accordingly. Host 10 typically begins sending data to the printer when the user selects a print command within an application program 11 running on host 10. Print commands from the application program 11 are passed through printer driver 12 to print spooler 13. Print spooler 13 temporarily stores plural print requests in local memory, and sequentially sends them to printer 15.

5.1 A7 > Except for real-time commands, which are further described below, printer 15 temporarily stores the sequentially sent and received data to receive buffer 17, which is a FIFO buffer. Data stored to receive buffer 17 is read in sequence from the first stored data, and any control commands are interpreted by data interpreter 18, which runs in CPU 16. Printer controller 19 controls driving the mechanical parts of the printer according to the interpreted control commands, and thus accomplishes printing and other mechanical operations. If the data interpreted by data interpreter 18 is a print command, CPU 16 expands the print data in receive buffer 17 in print buffer 20, and drives print head 21 and paper feed mechanism 22 appropriately to print and output the print data in print buffer 20.

15 The operation of printer 15 according to our invention can be set by a control command from host 10. That is, printer 15 has an area in RAM 23 reserved for storing these settings, and the specific values stored to this memory area can be set or changed as needed. The settings stored to RAM 23 are read as required and interpreted by CPU 16, which then performs the corresponding process.

5.1 A8 > These operation settings can also be stored to an EEPROM (electrically erasable programmable ROM), flash memory or other nonvolatile memory device (not shown in the figures) so that the stored content (settings) are not erased when printer 15 power is turned off.

5.1 A9 > Printer 15 has an area reserved in RAM 23 for setting the mode determining how to handle data received when the printer is off-line. This setting determines whether data already stored to receive buffer 17, data received while the printer is off-line, and data in print buffer 20 is held in memory or is destroyed when printer 15 is off-line. This setting is by default to hold the data, but by sending an appropriate command the host 10 can change the setting so that data is destroyed.

5.1 A10 > Printer 15 has a real-time command detector and processor 25 implemented in CPU 16 in conjunction with the control method of our invention. Real-time command detector and processor 25 detects whether received data is a real-time command before storing the received data to receive buffer 17. If the received data is detected to be a real, it immediately runs the process indicated by the real-time command before continuing or running any other process. As further described below, real-time commands are interpreted and run even when the off-line data processing mode is set to destroy data. It should be noted that font data, the various settings data, and a control program run by CPU 16 are stored to ROM 24. It will be obvious that ROM 24 can be plural physically discrete memory elements, or various kinds of rewritable nonvolatile memory such as an EEPROM device.

Printer 15 further has a status processor 26, also implemented by CPU 16. This status processor 26 provides an automatic status back (ASB) function for automatically informing host 10 about the printer 15 status independently of any request from host 10. To enable this ASB function, the host 10 first defines the printer status information that the ASB function is to monitor and report. When there is a change in any of this information, the printer 15 automatically notifies the host 10. By thus automatically receiving specific printer status data, host 10 can send print data and control printer 15 operation based on the ASB data without needing to request status information from the printer 15.

Host 10 relies on ASB reports from the printer 15 and real-time commands to monitor whether printer 15 is on-line or off-line.

Fig. 2 is a flow chart of the control process of our invention when printer 15 is off-line. The control process shown in Fig. 2 is performed when printer 15 goes off-line for a particular reason. Factors which could cause printer 15 to go off-line include, for example, the user opening the printer 15 cover, there being no paper loaded, or a specific error occurring in printer control.

CPU 16 detects when printer 15 goes off-line for some reason, and then starts the control process in Fig. 2. The first step 201 is to read and interpret the printer settings stored previously to RAM 23.

If the destroy data mode is set, all data in receive buffer 17 and print buffer 20 is cleared (step 202). One method for accomplishing this is a clearing unit that reinitialize the pointers used to specify the addresses in receive buffer 17 and print buffer 20. Alternatively, the printing motor may be disabled while data in receive buffer 17 continues to be processed. If the save mode is set, control advances to step 203 without clearing receive buffer 17 and print buffer 20.

Step 203 detects whether printer 15 is off-line.

If printer 15 has gone on-line again after this off-line control process has started because the user has switched printer 15 to the on-line mode, for example, control advances to step 215. Step 215 again detects the off-line mode setting. If set to the data destroy mode, receive buffer 17 and print buffer 20 are cleared (204) and the printer is then set to the on-line mode (205). If set to the data save mode, the printer is set to the on-line mode (205) without clearing receive buffer 17 and print buffer 20.

As further described below, if the destroy mode is set and data is not stored to the buffers while the printer is off-line, or if the destroy mode is set and data stored in the buffers continues to be processed and subsequently removed while the printer

is off-line and the printing motor is disabled, it is not always necessary to clear the buffers when the printer is returned on-line. If set to the save mode, however, clearing the buffers prevents errors resulting from data left in the buffer when the printer goes on-line.

5 It should be noted that printer 15 can receive data from host 10 while off-line.

If step 203 determines that the printer is still off-line, printer 15 awaits data from host 10 (206), and if data is received from host 10 while the printer is off-line, control advances to step 207. The real-time command detector and processor 25 then detects whether a real-time command is contained in the received data. If a
10 real-time command is not present, control advances to step 210.

If a real-time command is present, control advances to step 208 where the command is interpreted and run. Then control advances to step 210.

The off-line mode setting is again detected at step 210 to determine whether the save or destroy mode is set. If set to destroy, the data is destroyed without being saved to receive buffer 17 (211), and the procedure loops back to step 203 to repeat the process for any data received next. Alternatively in the destroy mode, the received data continues to be stored in receive buffer 17 and continues to be processed with the printing mechanism, i.e. printing motor, disabled or shut off. The loop from steps 203 to 210 is repeated for as long as printer 15 remains off-line. As a result, receive buffer 17 is prevented from becoming full while the printer is off-line, and the communication interface will not enter a busy state. Other method of destroying data during the off-line mode are likewise possible. For example, received data may be stored in received buffer 17 and not processed, but the receive buffer may be automatically cleared when the received data has filled a
15 predetermined amount of buffer space (i.e. 80% full) less than the amount that would typically result with a "buffer full" signal (i.e. 95% full).
20
25

If step 210 determines that the off-line save mode is set, control advances to step 212. Step 212 detects whether the receive buffer 17 is full. If not full, the received data is stored sequentially to receive buffer 17 (213). If the receive buffer
30 17 is full in step 212, the host 10 is notified that the interface is busy (214).

5.6 A.11 > Host 10 control relative to printer 15 control is described next. When the off-line printer mode is set to the data destroy mode in the flow chart shown in Fig. 2, all print data sent from the host 10 while the printer 15 is off-line is destroyed, as described above, and will not be printed. Control of the host 10 in this embodiment of the invention addresses this problem by a resending unit for resending print data
35 sent while the printer was off-line after the printer returns on-line. A CPU with a

program set stored in the ROM (and a component Q, if needed) may form the resending unit.

To detect whether all print data sent to printer 15 was printed by printer 15, host 10 uses a control command requesting notification that printing has been completed (referred to below as the printing completed command). This printing completed command is returned by printer 15 when the corresponding print data has been completely printed, and is sent in response to a request for such notification from host 10. In this exemplary embodiment of our invention the print data to which this command applies is the print data sent immediately before this printing completed command.

In a practical implementation of our invention this printing completed command can be achieved using a command having the same characteristics, that is, a command that is returned when printing is completed. For example, the printer control command ESC r1 can be used as this printing completed command. This ESC r1 command is returned to host 10 as a 1-byte status command identifying the status of paper in printer 15. If printer 15 is in the middle of printing, this ESC r1 command is sent to host 10 to confirm the paper status after printing is completed.

Fig. 3 is a flow chart of host 10 control when print data is sent to printer 15. The control process of the host 10 is described next below with reference to Fig. 3.

When printer 15 is confirmed to be on-line (301), host 10 sends print data with the printing completed command appended to the end thereof (302). After sending print data to the printer 15 with the printing completed command following the print data, host 10 waits for confirmation from the printer 15 that printing is completed (i.e., waits for a response corresponding to the above-noted printing completed command) (303). Host 10 continues to wait for this response as long as printer 15 does not go off-line (304). When the printer 15 completes printing and sends a response comparable to this printing completed command to host 10 (yes in step 303), host 10 determines that the transmitted print data was successfully printed and ends this process.

If printer 15 goes off-line due to some error or other problem before the printing completed command is returned, printer 15 reports to the host that it has gone off-line (304). When the host 10 receives this off-line report, control moves to step 305 to wait for printer 15 to return on-line. When the printer problem is corrected and the printer 15 returns on-line, host 10 receives an on-line report from the printer (305), and resends previously sent print data and printing completed

command (306). Note that this off-line and on-line status report from printer 15 is accomplished by the ASB function of printer 15.

If printer 15 is set to the data destroy mode while off-line in Fig. 2, any data in receive buffer 17 and print buffer 20 is destroyed when the printer 15 goes off-line. By sending print data to printer 15 according the control process shown in Fig. 3, print data sent by host 10 can be reliably printed irrespective of the off-line printer mode setting.

Fig. 4 is a flow chart of the printer 15 control process when the printer 15 is on-line and print data is sent from host 10 according to the process shown in Fig. 3. Printer 15 confirms whether data is stored to receive buffer 17 (401). If there is, the data is read in blocks of a specific size and passed to data interpreter 18 (402). The data interpreter 18 detects whether the read data is print data (403), and if it is CPU 16 runs the printing process (404). Step 404 repeats until all print data has been printed.

When all print data read in step 402 is processed, the printing completed command appended to the print data will be read next, and step 403 will pass control to step 405. When the read data is the printing completed command, step 405 waits for the printing process to be completed (406) and then sends a response corresponding to the printing completed command to host 10 (407). When the host 10 receives this response, it confirms that all sent print data has been successfully printed as described above. If printer 15 goes off-line due to an error or other reason before printing is completed (408), the off-line control process (409) shown in Fig. 2 is run.

Although the present invention has been described in connection with the preferred embodiments thereof with reference to the accompanying drawings, it is to be noted that various changes and modifications will be apparent to those skilled in the art. Such changes and modifications are to be understood as included within the scope of the present invention as defined by the appended claims, unless they depart therefrom.

For example, the invention shall not be limited to the data transmission procedure of the host 10 described above, and includes procedures for sending data according to other known methods.

Furthermore, this exemplary embodiment has been described as destroying all data other than real-time commands sent from host 10 when printer 15 is off-line and set to the data destroy mode. However, it is alternatively possible to clear receive buffer 17 and print buffer 20 when the printer goes off-line and then store

data received while the printer is off-line. In this case the mode setting is a 2-bit command, one bit used to set the data handling procedure when the printer enters the off-line mode, and one bit used to set the data handling procedure while the printer remains off-line. An optional third bit can indicate whether stored data should continue to be processed while the printing motor is disabled or shut off.

Furthermore, because a busy signal is sent to host 10 when receive buffer 17 becomes full, it is also possible to clear only receive buffer 17 and not clear print buffer 20 when off-line.

As will be known from the above description, our invention prevents the printer receive buffer from becoming full and the communication interface with the host computer from entering a busy state even when a large volume of data is received while the printer is off-line. The host computer can therefore continue to send control commands to the printer, and can therefore determine the printer status or recover the printer from a recoverable error state. In addition, because data is destroyed by the printer and thus prevents the communication interface from returning a busy signal due to a full receive buffer, the print spooler is prevented from generating an error in systems in which the host computer sends print data to the printer via a print spooler.

It is also possible to selectively set a printer according to our invention to save or destroy received data when the printer is off-line, and the printer can thus be set to the mode appropriate to the current operating conditions.

Furthermore, real-time commands can still be executed when the printer is off-line, thereby enabling printer status to be determined and recoverable printer errors to be corrected using such real-time commands.

Yet further, the host computer can resend print data when data in the receive buffer has been cleared by the control method of our invention, and print data can therefore be reliably printed.

While the invention has been described in conjunction with several specific embodiments, it is evident to those skilled in the art that many further alternatives, modifications and variations will be apparent in light of the foregoing description. Thus, the invention described herein is intended to embrace all such alternatives, modifications, applications and variations as may fall within the spirit and scope of the appended claims.

Still further, while the present invention has been described as controlling a printer depending on whether it is in the on-line state or the off-line state, the same principle may be applied to other states. The "off-line state" has been defined as a

state in which the printer does not interpret received data in the receive buffer. For printers that do not print, but do interpret data when the printer is off-line the above definition of the term "off-line" would not be correct. For such types of printer, for instance, the "off-line state" in the preceding description may be replaced by "first state" defined as a state in which the printer does not print received data, while the "on-line state" may be replaced by "second state" defined as a state in which the printer prints received data.

KEY TO THE FIGURES

- | | | |
|----|----|------------------------------------------|
| | 10 | host computer |
| 10 | 11 | application program |
| | 12 | printer driver |
| | 13 | print spooler |
| | 14 | interface cable |
| | 15 | printer |
| 15 | 16 | CPU |
| | 17 | receive buffer |
| | 18 | data interpreter |
| | 19 | printer controller |
| | 20 | print buffer |
| 20 | 21 | print head |
| | 22 | paper feed mechanism |
| | 23 | RAM |
| | 24 | ROM |
| | 25 | real-time command detector and processor |
| 25 | 26 | status processor |